AWS Security Blog

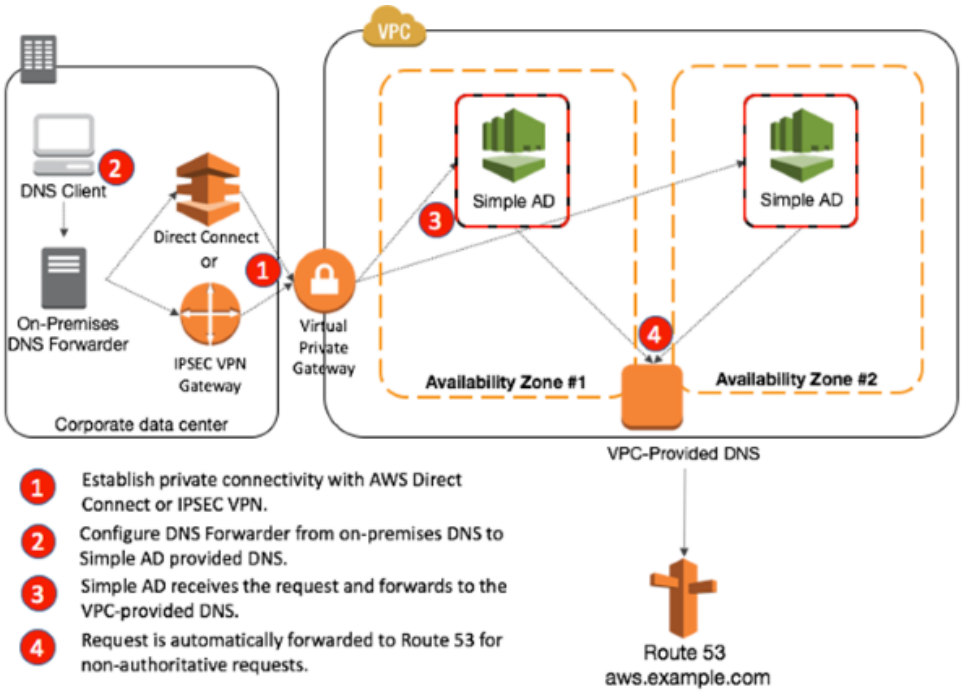
How to Set Up DNS Resolution Between On-Premises Networks and AWS Using AWS Directory Service and Amazon Route 53

by Drew Dennis | on 01 FEB 2016 | in Amazon Route 53, AWS Directory Service, How-To | Permalink | Comments | Share

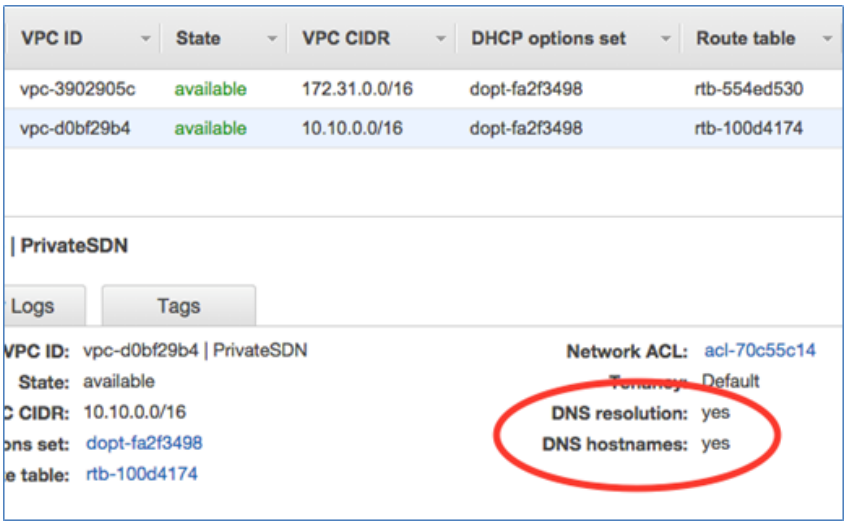
As you establish private connectivity between your on-premises networks and your AWS Virtual Private Cloud (VPC) environments, the need for Domain Name System (DNS) resolution across these environments grows in importance. One common approach used to address this need is to run DNS servers on Amazon EC2 across multiple Availability Zones (AZs) and integrate them with private on-premises DNS domains. In many cases, though, a managed private DNS service (accessible outside of a VPC) with less administrative overhead is advantageous. In this blog post, I will show you two approaches that use Amazon Route 53 and AWS Directory Service to provide DNS resolution between on-premises networks and AWS VPC environments.

Using AWS Directory Service Simple AD to forward DNS requests to Route 53

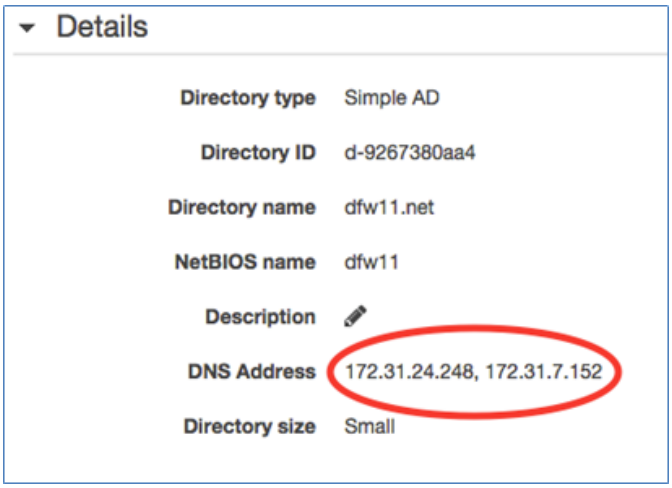
Simple AD provides redundant and managed DNS services across AZs. These DNS services automatically forward requests for non-authoritative domains to the VPC-provided DNS. Therefore, they can be used to resolve DNS records stored in a Route 53 private hosted zone. The following diagram shows this architecture.



For example, if you provision a Simple AD directory and give it a name of `example.net`, any DNS request outside of that domain name (let's assume aws.example.com) is forwarded to the internal DNS service of the VPC. If a Route 53 private hosted zone has been created for aws.example.com and assigned to that VPC, it responds to the DNS queries that originate from outside the VPC. This is accomplished by way of the Simple AD DNS service that integrates DNS resolution across Simple AD resources, VPC-provided DNS, and Route 53 private hosted zones. Note that the VPC needs to have DNS resolution and DNS hostnames enabled, as shown in the following screenshot of the VPC console. See Using DNS with Your VPC for more details about these settings. For additional details about the DNS service provided with AWS Directory Service, see Using DNS with Simple AD and Microsoft AD.



After the Simple AD directory has been provisioned, selecting it from the AWS Directory Service console displays two associated IP addresses for DNS, as shown in the following screenshot.



Any request sent to these IP addresses is forwarded to the VPC-provided DNS service and Route 53. Setting up DNS forwarders in your on-premises DNS service to these IP addresses for your Route 53 domain names is an easy way to realize immediate DNS resolution from on-premises hosts into your AWS VPC. Be sure to check the security group created for your directory to ensure DNS traffic is allowed from your on-premises networks. Also, make sure the Route 53 domain name is different than the Simple AD domain name. If they are the same or if the Route 53 domain is a subdomain of the Simple AD domain, Simple AD does not forward the request.

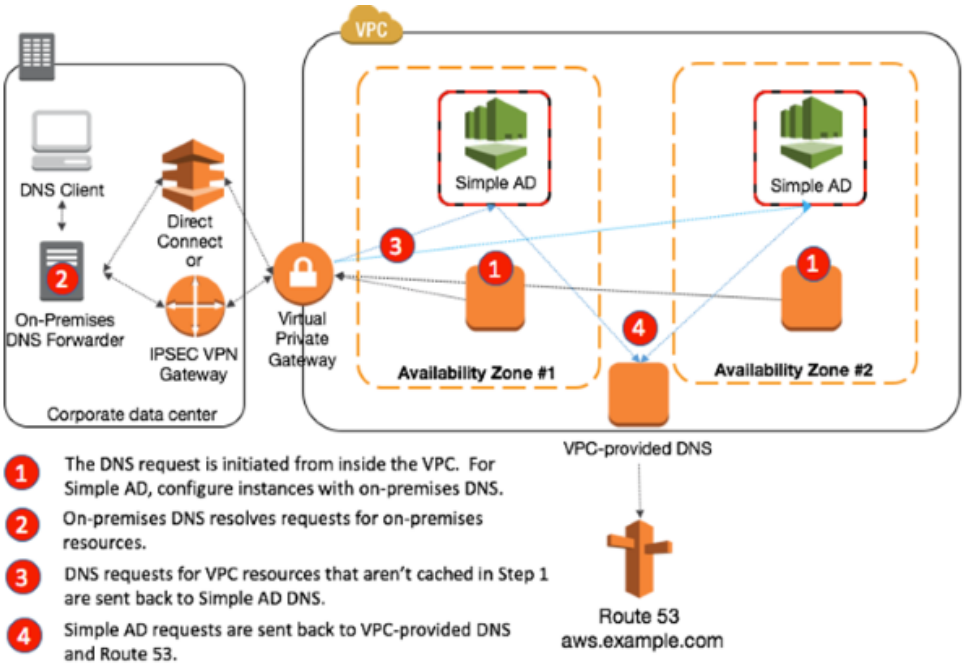
Because Route 53 is a global service and its private hosted zone can be associated with multiple VPCs, this scenario can provide DNS resolution for hosts and services located in multiple VPCs across multiple AWS regions. This enables you to have multiple VPCs all using a single centralized Route 53 private hosted zone.

Resolving DNS requests for on-premises resources originating from AWS

In most cases, you also want resources that are deployed inside your VPCs to be able to resolve names for resources that exist in your data centers or on-premises networks. In other words, the steps I have covered so far in this post have been about resolving VPC resources from existing corporate or private networks. How about resolution in the opposite direction?

If Simple AD is the directory service of choice, routing DNS requests back through on-premises DNS servers can be the best option. I have already covered the steps to forward those requests to the VPC and Route 53. Configuring EC2 instances in your VPC with the IP address of an on-premises DNS server (with a forwarder to Simple AD's DNS server) can provide a single DNS configuration that can resolve names, both within AWS and on your private networks. VPC DHCP option sets can be leveraged to automate this. See DHCP Option Sets for more details.

The following diagram depicts this flow of DNS requests originating from inside the VPC with Simple AD. Only instances that need on-premises DNS resolution should be configured with an on-premises DNS server. If an instance needs only to resolve names for VPC resources, you should configure it with the address of the Simple AD–provided DNS.



Summary

AWS Directory Service can make integration between on-premises and AWS-managed DNS services easier and can eliminate the need to manage your own DNS servers. You can use the Simple AD directory type to forward DNS requests that originate from on-premises networks to the VPC-provided DNS and ultimately Route 53. You might find flexibility by running DNS services on EC2, but that adds additional management overhead and needs to be configured for high availability and failover. The Directory Service–provided DNS for Simple AD provides these services across AZs as part of the provisioned directory. This can make implementing hybrid architectures on AWS much easier.

In my next post, I will take a look at using Microsoft Active Directory instead of Simple AD to provide DNS resolution for hybrid architectures. If you have questions or comments, submit them below or on the Directory Service forum.

– Drew

TAGS: Amazon Route 53, AWS Directory Service, AWS VPC, DNS, DNS resolution

Related Posts

Enhanced Domain Protections for Amazon CloudFront Requests

New .BOT gTLD from Amazon

Use a custom domain with AWS AppSync, Amazon CloudFront, and Amazon Route 53

Amazon ECS Service Discovery

How to Use Bucket Policies and Apply Defense-in-Depth to Help Secure Your Amazon S3 Data

Combining DynamoDB and Amazon Elasticsearch with Lambda

Dynamically Create Friendly URLs for Your Amazon EMR Web Interfaces

How to get started on AWS (From a dead standstill)